

4. Documentation Technique

Documentation Technique – Esportify

1. Langages & Technologies

- **Front-end** : HTML5, CSS3 (Bootstrap partiel), JS Vanilla
- **Back-end** : PHP (PDO), requêtes SQL sécurisées
- **Base de données** : MariaDB (tables : users, events, inscriptions, scores, galerie, chat)

2. Architecture

- **/front/** : pages HTML, scripts JS (séparés par fonctionnalité), assets (images, logos, etc.)
- **/back/** : scripts PHP (connexion base, API AJAX, contrôleurs de chaque action)
- **Organisation logique** : chaque fonctionnalité (auth, gestion événements, inscription, modération, chat) dispose de ses propres fichiers côté back et front.

3. Sécurité

- **Sessions PHP** sécurisées (stockage login, rôles, vérification accès aux pages sensibles)
- **Mots de passe hashés** (password_hash)
- **Requêtes SQL sécurisées** (requêtes préparées via PDO, protection contre injections)
- **Validation systématique** des entrées utilisateurs côté serveur et côté client (JS)

4. Modèle de données

- **users**
 - id (PK)
 - pseudo
 - email
 - mot_de_passe (hashé)
 - role (joueur, organisateur, admin)
- **events**
 - id (PK)
 - titre
 - description
 - date_event, heure_event
 - jeu
 - statut (en attente, validé, refusé)
 - id_createur (FK)
 - max_joueurs
 - image_url
- **inscriptions**
 - id
 - user_id (FK)
 - event_id (FK)
 - date_inscription
- **scores**
 - id
 - user_id (FK)
 - event_id (FK)
 - points
 - classement

5. Séparation des rôles

- **Joueur** : s'inscrire à un événement, accéder au chat. Un joueur dont la proposition d'événement est validé par l'admin devient alors Organisateur.
- **Organisateur** : créer/modifier événements, gérer participants.
- **Admin** : valider/refuser événements, gérer utilisateurs et droits, modérer chat.

6. JS/Front

- **dashboard.js** : gestion du tableau de bord (chargement dynamique, actions, modales).
- **event-repo.js** : listing dynamique, filtres, inscription AJAX.
- **main.js** : fonctions globales (navigation, modales, chat, classement).
- **navbar.js** : affichage dynamique du menu selon le rôle/session.
- **galerie.js** : gestion carrousel, upload/suppression images (admin).
- **responsive.css** : media queries pour l'adaptation mobile/tablette/desktop.
- **login.js** : gestion connexion AJAX.

7. Déploiement

- **Déploiement** sur AlwaysData (serveur mutualisé).
- **Dépôt GitHub** public avec branche main et dev :
 - Une branche principale (main)
 - Une branche de développement (dev)
 - Branches de fonctionnalité, merge vers dev puis vers main après validation.
- **Fichier README.md** pour le guide d'installation locale.

8. Sécurité et RGPD

- Pas de collecte de données personnelles en dehors des besoins du projet.
- Données non exposées en clair dans le front.
- Administration protégée.
- Respect du RGPD (possibilité de supprimer compte, pas d'usage commercial).